Computer Applications in Experimental Physics

I. OBJECTIVE 1
II. INTRODUCTION 1
III. EXERCISES 1
A. Preliminaries 1
B. Taking Your Own Data via a GPIB Port 1
C. Analyzing Your Data 2
IV. APPENDIX 1: QUICKBASIC PROGRAM FOR READING KEITHLEY MULTIMETER 3

I. Objective

To understand the hardware and software involved in using computers in experimental physics applications and to perform an experiment controlled by a computer.

II. Introduction

The goal of Physics 232 is to present the various uses of computers in physics. Up to now we have been concerned with numerical solutions to mathematical problems using different techniques (spreadsheets, symbolic math packages, programming). Today we will focus on how computers may be used to acquire and analyze data. We will perform a lab to emphasize the key points by writing a program in **QuickBasic** to take data from a Keithley multimeter and study the heating of ice water. You will also use Excel to analyze the data by performing a least squares fit. Next week (4/11/97), we will have a lab tour to see how the research groups in the department use computers.

III. Exercises

A. Preliminaries

If you took the course PHY 222 at ISU during a recent semester, you performed lab No. 12 **The Computer as a Laboratory Instrument** . This lab introduced a variety of ways to interface an experiment with a computer: **BASIC** as an easy-to-use language (particularly for controlling experiments), **parallel input** and **parallel output**, **relais, timers, counters**, **GPIB-bus** etc. Please have your PHY 222 lab handout ready for an introduction to BASIC and a description of the GPIB interface. A more thorough description of the GPIB bus can be found in a series of articles in the journal **Computers in Physics** , see the PHY 232 homepage.

B. Taking Your Own Data via a GPIB Port

Objective:	to write a QuickBasic program to take data from a Keithley multimeter through a GPIB port
Where to begin:	in room 82 (Physics 222 Lab), page 27 of Lab #12
What to do:	follow the instructions below
What to turn in to your instructor:	your log book, a copy of your QuickBasic program
What to put in log book:	time you begin/end your work, problems encountered, solutions developed, interesting facts.

(1) Starting QuickBasic: To open the QuickBasic programming environment, type the command **qb** at the **C:\>** prompt. You should see a blue window with the standard menus across the top.

(2) Loading and Running Program to Read Multimeter: Parts of what we will do in today's lab you did in Physics 222 during a lab called *The Computer as a Lab Instrument* (Lab #12). In case you do not have a copy of this lab with you, there should be a desk copy of the lab description at your lab table. For an overview of QuickBasic, see pages 11-20 of Lab #12.

Turn to page 27 in Lab #12, follow the instruction there, and record necessary information in your log book. Work through sections V. and VI. of the lab. A simple version of a QuickBasic program designed to read data from the multimeter is given on page 28 along with an explanation of the code.

The program on page 28 is stored in **C:\CLI\SAMPLE1.BAS**, which you may load using the command **Open Program** in the **File** menu. To run the program, select the **Run** menu and the command **Start.** You should find that the resistance reading from the multimeter is printed out on the screen without a decimal point.

(3) Modifying the Program: Modify the program to perform the following operations:

- Read resistance values from the multimeter in a loop until the key **Q** is pressed. Note that because the decimal point is not read, you will have to make the program divide the resistance reading by an appropriate power of ten.
- Convert the resistance values into Celsius temperature values. Page 31 of Lab #12 shows that the conversion formula for resistance values to Celsius temperatures is given by

$$T(C) = \left(\frac{100-0}{138.5-100.0}\right)(R-100)$$

  Show this derivation in your log book.
- Write the time and the temperature to the screen and also into a file called **data.dat**

If you have questions about this, see Appendix 1 for an example.

(4) Taking Data: When your program is running properly, obtain a cup of ice. Begin your program with the resistance thermometer at room temperature. With the program running, place the thermometer in a cup of ice water. Continue to take data until the temperature stabilizes. When you are done, copy your data files to a floppy disk for further analysis.

C. Analyzing Your Data

Objective:	to perform a least squares fit using Excel
Where to begin:	in Excel
What to do:	follow the instructions below
What to turn in to your instructor:	your log book, a copy of your Excel graph
What to put in log book:	time you begin/end your work, problems encountered, solutions developed, interesting facts.

(1) Experimental Model: According to Newton's Law of Cooling, an object looses heat at a rate proportional to the difference between its temperature and the temperature of the reservoir it is in contact with. If the reservoir temperature is T_R and the initial temperature is T_0, then the temperature at any time t is given by

$$T(t) = (T_0 - T_R)e^{-Ct} + T_R$$

We will fit the data taken in the experiment in part B to the above formula using the method of least squares.

(2) Least Squares Fitting: In the method of least squares, one tries to minimize the quantity Δ defined as

$$\Delta \equiv \frac{1}{N}\sum_{i=1}^{N}\Delta_i$$

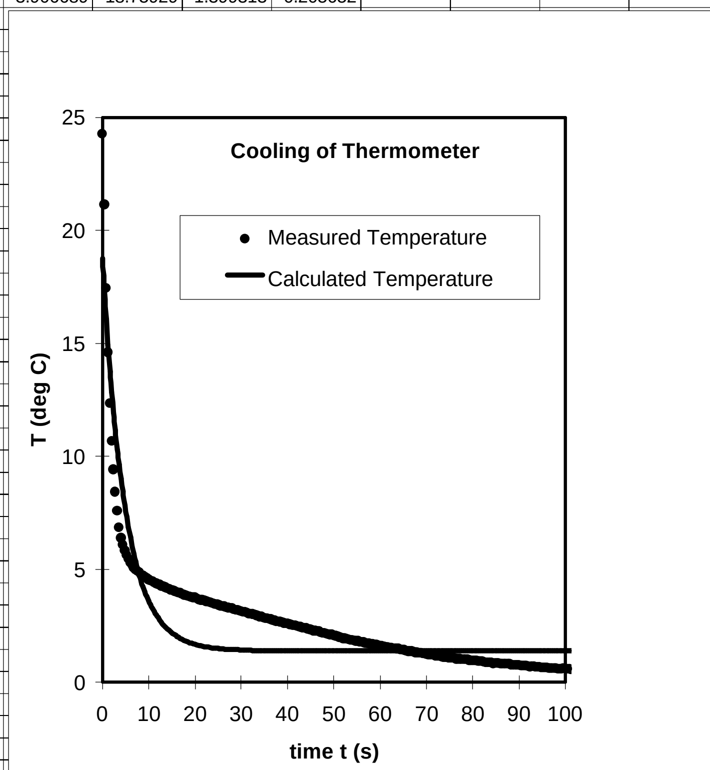
where $\ _i \equiv (T_i^{exp} - T_i^{theor})^2$, T_i^{exp} and T_i^{theor} represent respectively the experimental and theoretical values of the measured quantity at data point *i* taken at time t_i, and the summation is taken over all *N* data points. Therefore, Δ measures how much the theoretical curve deviates from the experimental curve over the entire data range. Minimizing Δ is one criterion for selecting good fits to experimental data.

(3) Importing Data into Excel: Import the data you took in part B into your Excel spreadsheet by using the **Open** command under the **File** menu. Insert a few extra rows at the top of your data.

Set up your spread sheet as indicated below, with cells at the top containing parameters T_0, T_R, and *C*. The value of T_0 is determined by your initial temperature value, and that of T_R may be assumed to be 0^o C for the ice water bath. The parameter *C* is as yet unknown, so make a guess.

Create two additional columns, one for the theoretical temperature T_i^{theor} value expected from Newton's Law of Cooling above (this must reference the cells containing T_0, T_R, and *C*), and one for the square of the difference $\ _i \equiv (T_i^{exp} - T_i^{theor})^2$. Next, create a cell at the top of the spread sheet called **sumdelta** that contains the sum $\Delta \equiv \frac{1}{N}\sum_{i=1}^{N}\Delta_i$, and set up a graph of T_i^{exp} and T_i^{theor} as shown. Try to improve your fit by adjusting *C* by hand. (In the next step, we will use Excel to optimize the values of the parameters.)

Time t	Temp T	Time t	T (calc)	Delta T	Sum Delta	T0	TR	C
s	deg C	s	deg C	deg C	3.906689	18.75929	1.399313	0.205652
0	24.28571	-2.85938						
0.0625	24.28571	-2.79688						
0.5	24.25973	-2.35938						
0.878906	24.28571	-1.98047						
1.269531	24.28571	-1.58984						
1.710938	24.25973	-1.14844						
2.089844	24.25973	-0.76953						
2.46875	24.25973	-0.39063						
2.859375	24.25973	0	18.75929	30.25489				
3.300781	21.14285	0.441406	17.25282	15.13231				
3.679688	17.45455	0.820313	16.06437	1.932595				
4.070313	14.59741	1.210938	14.93236	0.112192				
4.511719	12.33766	1.652344	13.75799	2.01734				
4.890625	10.67533	2.03125	12.83153	4.649203				
5.28125	9.428564	2.421875	11.94907	6.352934				
5.660156	8.415579	2.800781	11.15821	7.522034				
6.101563	7.584411	3.242188	10.31135	7.43621				
6.480469	6.857141	3.621094	9.643268	7.762501				
6.871094	6.389608	4.011719	9.006908	6.850262				
7.25	6.077912	4.390625	8.43661	5.563458				
7.691406	5.844156	4.832031	7.825928	3.92742				
8.082031	5.636359	5.222656	7.329851	2.867915				
8.460938	5.454542	5.601563	6.885271	2.046986				
8.902344	5.298704	6.042969	6.409211	1.233226				
9.289063	5.194805	6.429688	6.026207	0.691229				
9.671875	5.064927	6.8125	5.675918	0.373311				
10.05078	4.987008	7.191405	5.355327	0.135659				
10.49219	4.909089	7.632815	5.012029	0.010597				
10.87891	4.85715	8.019535	4.735838	0.014717				
11.26172	4.753252	8.402345	4.483241	0.072906				
11.69922	4.701293	8.839845	4.217887	0.233681				
12.08984	4.649353	9.230465	4.000321	0.421243				
12.46875	4.597394	9.609375	3.805336	0.627355				
12.85938	4.545455	10.00001	3.619611	0.857188				
13.28906	4.493515	10.42969	3.431833	1.127168				
13.67969	4.467535	10.82032	3.274939	1.422285				



(4) Performing the Least Squares Fit: Suppose we wish now to minimize the variable **sumdelta** by varying the parameter **C**. This is a one parameter least squares fit.

To perform the minimization of **sumdelta**, we will use a function called **Solver.** Select the **Tools** menu and the **Solver** menu item. A box will open containing several prompts. Locate the prompt **Select target cell** and specify the cell address of **sumdelta,** the quantity we desire to minimize.

Next locate the prompt **Equal to** and select the minimization button **min**. Finally locate the prompt **By changing cells** and select the cell location of the parameters to be adjusted, in this case **C.**

After the above selections have been made, begin the minimization by clicking the button **Solve.** You should see the iterations begin. After several iterations, Excel will modify the graph of T_i^{theor} for the new value of *C* that minimized the sum of the squares. Record this value of *C* in your log book and print out a copy of the graph showing the data and the fit to the data.

(5) Playing with the Fit: Now try fitting with two and three parameters. Are the results more or less consistent with what you expect? Report and discuss your findings in your log book. Generate the three reports that the solver can create. Do they contain any kind of useful information? What information would you like to have which is not contained in these reports?

(6) Marquardt-Levenberg algorithm : How does the Excel solver work? I have no idea. Unfortunately, the documentation does not tell us. Nonlinear least squares fitting is typically performed using the Marquardt-Levenberg algorithm described in **Numerical Recipes**. Standard software libraries (such as NAG or IMSL) typically have routines for this algorithm, therefore you do not need to know a lot more about it than its name. This algorithm can also tell you the **errors of the parameters** and whether your problem is overdetermined (i.e., contains more parameters than needed to describe the data). **Sigmaplot** by Jandel Scientific contains a nice Marquardt-Levenberg Solver.

IV. Appendix 1: QuickBasic Program for Reading Keithley Multimeter

Listed below is one way to write a program that will read resistance values from a Keithley Model 177 Multimeter. The program writes the time and temperature to the screen and to the file **C:\CLI\DATA.DAT**. The program continues to take data until the **Q** key is pressed.

```
7000  'Reading a KEITHLEY model 177 multimeter via the IEEE-488 interface
7005  CLS
7010  OPEN "GPIB0" FOR OUTPUT AS #1     'PREPARE GPIB #0 FOR OUTPUT TO METER
7020  OPEN "GPIB0" FOR INPUT AS #2      'PREPARE GPIB #0 FOR INPUT FROM METER
7025  OPEN "C:\CLI\DATA.DAT" FOR OUTPUT AS #3  'OPENS DATA FILE
7026  INIT = TIMER                      ‘GETS INITIAL TIME IN SEC
7030  PRINT #1, "ENTER 12#8"            'TELL METER AT ADDRESS 12 TO PRESENT A
                                        ‘READING OF 8 CHARACTERS
7040  INPUT #2, R$                      'ACCEPT READING INTO MEMORY RESERVED FOR R$
7050  R = VAL(R$) / 100                 'CONVERT CHARACTERS INTO A NUMBER
7055  C = 100 / 38.5 * (R - 100)        'CALCULATES CELCIUS TEMPERATURE
7060  PRINT TIMER - INIT, C             'PRINTS TIME(SEC) AND TEMP(C) TO SCREEN
7063  PRINT #3, TIMER - INIT, C         ‘PRINTS TO FILE C:\CLI\DATA.DAT
7065  IF INKEY$ = "Q" THEN GOTO 7080    'TERMINATES IF Q KEY PRESSED
7070  GOTO 7030                         'LOOPS BACK FOR ANOTHER DATA POINT
7080  END
```

V. Appendix II: Modern (Protected) Operating Systems:

DOS (the operating system which you are running on this computer) was designed to run one program, i.e., execute a series of instructions sequentially. More sophisticated operating systems (such as Windows NT, OS/2, UNIX, or OS-9) can run several **processes** (for example one for each user logged on to the computer). Each of the processes can have more than one **thread**.

These operating systems are usually **protected** in the following sense: User A is not supposed to read or write to the memory belonging to user B. It also makes sense that only one process can read or write to a physical port (for example, the parallel port or the GPIB interface) at a time. For this reason, port I/O is a protected command on modern operating system. An attempt to execute a **port I/O** command will give you an error message an stop the process.

In order to allow port I/O and define a hierarchy of commands, the Intel 386 (and above) chip has a sophisticated **ring structure**. Normal (user) processes execute in **ring 3**, where port I/O is forbidden. In order to do port I/O, a program needs to call a **ring 2 segment**, where port I/O is allowed. The operating system itself runs in **ring 0**, where any command is allowed. See my article in Computers in Physics (link at the PHY 232 Web site) for more details about how to port I/O in OS/2 or Windows NT. This ring structure is not implemented in DOS or Windows 3.1. (I am not sure about Windows 95.) These operating systems merely use the ancient 80286 chip architecture.